Sensory Channel Grouping and Structure from Uninterpreted Sensor Data

Lars Olsson, Chrystopher L. Nehaniv, and Daniel Polani
Adaptive Systems Research Group
Department of Computer Science
University of Hertfordshire
College Lane, Hatfield Herts AL10 9AB
United Kingdom
{L.A.Olsson, C.L.Nehaniv, D.Polani}@herts.ac.uk
+44(0)1707-284000

Abstract

In this paper we focus on the problem of making a model of the sensory apparatus from raw uninterpreted sensory data as defined by Pierce and Kuipers (Artificial Intelligence 92:169-227, 1997). The method relies on generic properties of the agent's world such as piecewise smooth effects of movement on sensory features. We extend a previously described algorithm with an information-theoretic distance metric that can find informational structure not found by the original algorithm. We also use the method to create metric projections of the sensory and motor systems of a robot. Data from a real robot show that the metric projections for example can be used to distinguish the vision sensors from all other sensors and also to find their functional layout. Finally we present an application of the method where the real layout of the vision sensors is found from scrambled vision data.

1. Introduction

> When Gregor Samsa awoke one morning from troubled dreams he found himself transformed in his bed into a monstrous bug.
>
> Franz Kafka, *The Transformation*

Faced with the situation of having his brain "plugged in" to the body of a monstrous bug, the main character of Franz Kafka's novel *The Transformation* [9] begins the story with struggling to understand how to control his newly gained body with its numerous legs and new sensory input. The ability to understand how different sensors are related to each other, the outside world, and the motoric system, is not advantageous only to the bizarre scenario of people that have been transformed into bugs but many man-made systems as well. For example, consider an autonomous robot on the surface of Mars that is hit by some space debris. Maybe parts of the central processing unit are damaged and need to be reconfigured [13], or the camera and lens might be displaced or damaged. In both situations there is a need for the control system to learn a new model of the robot's sensors. Another example of the need to organize sensory channels arises from the fact that modern sensors provide high data rates where not all of the information is relevant to solve a certain task, while storage and on-board processing capabilities are limited [11]. Also, in some situations, e.g., space exploration, not everything about the environment might be known and the conditions, like for instance light and temperature, can change drastically over time. Hence the designer or the system itself needs to adapt the sensory apparatus over time or after deployment to select the information most relevant to perform its particular tasks. The notion of relevant information was introduced in [14] and formalized in [17] by associating the relevance of information with the utility to an agent to perform a certain action. Given knowledge of the most relevant information from a number of sensors it might then be possible to adapt the sensoric system to discriminate only between events that are of use for the system.

Adaptive sensory organization is displayed over evolutionary time in nature where we find a wide variety of sensory organs that are well adapted to the specific animals and their respective environment [6]. Information theoretic properties can yield insight into the evolution of sensors and actuators [14]. Evolution can also be used in artificial systems [5] to find better configurations of existing sensors or even finding new "meta-sensors" [2]. This might for example be realized using evolvable hardware [8].

In this paper we focus on the problem of making a model of the sensors from raw and uninterpreted data as defined by Pierce and Kuipers in [15]. By raw uninterpreted data we mean multiple time series of data where the sources of the information (the sensors) are not known, and it is the task of the system to group different sensory channels together. Usually it is the designer of a robot that specifies the functional relationships between sensors, the physical body of the robot, the control system, and the actuators. In this paper these relationships are not known.

In [15] Pierce and Kuipers describe how a simulated agent can learn a spatial model of the sensory apparatus from raw uninterpreted sensory data in a static environment. The learning methods rely on generic properties of the agent's world such as piecewise smooth effects of movement on sensory features. The results of the learning algorithm is a set of groups of related sensors and a spatialized description of the layout and dimensionality of the sets of sensors.

The work presented in our paper extends the algorithm by Pierce and Kuipers by introducing a method for finding informational structure in the uninterpreted sensory data. Moreover, we give the first implementation of this method on real world data. The method uses the information distance, which was defined and shown to satisfy the axioms for a metric in [4][1]. The distance is between two information sources in the sense of classical information theory [18] in terms of conditional entropies. Our results show that the information metric can be used to find informational structures not found by the original algorithm as described in [15]. We also consider data from the physical world by presenting data from experiments with a robot (SONY AIBO[2]) that actively moves around in the world. Here we utilise the method created by Pierce and Kuipers and extended by the information metric to create metric projections of the sensory input and motor control, in a sense analogous to sensory-motor maps. The results indicate that these maps can be useful to understand the relationships between different sensors and the motor control. In the last experiment a metric projection of a number of vision sensors is used to recover the layout of a vision system that seems to produce noise.

The structure of this paper is as follows. Section 2 describes the used algorithms and Section 3 the experiments and their results. The first experiment uses artificial data to show different properties of the distances metrics described in Section 2. In the next experiment real world data is used to show the possibility of creating metric projections of sensors and motoric systems. The final experiment presents an application of the metric projections to recover structure in scrambled image data. Finally Section 4 summarizes the paper and consider future applications, like sensor networks.

[1] Formally, a metric requires (1) $d(X,Y) = d(Y,X)$, (2) $d(X,Y) = 0$ iff $Y = X$, and (3) $d(X,Y) + d(Y,Z) \geq d(X,Z)$. If (2) fails but (1) and (3) hold, then we have a pseudo-metric, from which one canonically obtains a metric by indentifying points at distance zero from each other. This is done here and in [4].

[2] AIBO is a registered trademark of Sony Corporation.

2. Sensory Reconstruction Method

This Section describes the sensory reconstruction method described by Pierce and Kuipers [15]. The first four sections describe Pierce and Kuipers's algorithm, while Section 2.5 discusses the informational metric between sensors.

2.1. Input Data

The input data to the sensory reconstruction method is a number of time series of sensory data, one for each sensor x. Each element in each time series is a real value number normalized in the range $[0.0, 1.0]$. Thus, there are n time series $x_1, x_2, .., x_n$, one for each sensor, each t time steps long. The number of time steps t needed depends on the number of sensors and their characteristics, but in general the more complex the environment and the sensors, the more data is needed to build a good model.

2.2. Distance Metrics

Given the sensors $x_1, x_2, .., x_n$ with time series of data the distance between each sensor to every other sensor is computed using two different distance metrics, the normalized *Hamming metric*, $d_1(x_i, x_j)$ and the *frequency metric*, $d_2(x_i, x_j)$.

The first measure is based on the idea that two sensors x_1 and x_2 that have similar output over time are similar. The Hamming metric is defined as

$$d_1(x_i, x_j) = \frac{1}{t+1} \sum_{\tau=0}^{t} |x_i(\tau) - x_j(\tau)| \qquad (1)$$

where τ is a time index between 0 and t, the length of the time series. The result is a value in the range $[0.0, 1.0]$ where 0.0 means that the sensors are identical and 1.0 that they are maximally different.

The second metric, the frequency metric, measures the similarity of the frequency distributions of the two sensors. Given a sensor x, we can create a vector of n elements, $distr(x)$, where each element specifies in percent how often the sensor has a value in a particular subinterval. The frequency metric is defined as

$$d_2(x_i, x_j) = \frac{1}{2} \sum_{\ell=1}^{n} |distr(x_i)_\ell - distr(x_j)_\ell|, \qquad (2)$$

where $distr(x_i)_\ell$ is the percent of observations within the ℓth subinterval. This metric again results in a value in the range $[0.0, 1.0]$, where 0.0 means that the sensors are completely similar with regards to the frequency distributions. A value of 1.0 means that the distributions do not overlap at all.

2.3. Grouping

Given the distances according to a metric d between all sensors it is possible to form subgroups of related sensors in the following way. Let sensor x_i and x_j be similar, written

$$x_i \approx x_j \quad \text{if} \quad d(x_i, x_j) \leq \min\{\epsilon_i, \epsilon_j\} \tag{3}$$

where the ϵ's are thresholds. In [15] each sensor x_i has its own threshold ϵ_i calculated from the minimum distance to any of its neighbours:

$$\epsilon_i = 2 \min_j \{d(x_i, x_j)\} \tag{4}$$

To form closed subgroups [15] use the *related-to* relation which is the transitive closure of the $\approx$ relation. This is computed recursively via

$$i \sim j \quad \text{iff} \quad i \approx j \vee \exists k : (i \sim k) \wedge (k \sim j). \tag{5}$$

2.4. Dimensionality and Structure

Given a closed subgroup of sensors, it is possible to find their dimensionality and structure using the difference matrix of $d_k(x_i, x_j)$. The first step is to find the dimensionality of the group of sensors. This can be done by considering the amount of variance in the sensor data that is accounted for by each dimension m. Let $\sigma^2(m)$ be the variance accounted for by dimension m. Then the right number of dimensions can be found by maximizing $\sigma^2(m) - \sigma^2(m+1)$. This can be computed using a scree-diagram [15]. For example, if each sensor is a pixel in a camera this would result in $m = 2$.

Given the dimension one can then compute a metric projection of the sensors by applying for example metric scaling [12] to the difference matrix of d_k. A smoother projection with more evenly laid out points can be computed using the relaxation algorithm of [15].

Goodhill and Sejnowski [7] discuss a number of other methods like Sammon mapping and elastic nets that can be used to create metric projections instead of metric scaling and relaxation. Another method worth investigating is Self-Organizing Maps [16]. As we will see in the experiments in Section 3 it can also be rewarding to create a metric projection of all sensoric input and not just the ones in a closed subgroup using different distance metrics.

2.5. Information Distance between Sensors

After a reimplementation of the sensor reconstruction method of Pierce and Kuipers [15] we felt that the Hamming metric and the frequency metric were not sufficient to capture all informational relations for the problems we were interested in. As we will see later in the paper related processes may exhibit large distances in the Hamming metric and the frequency metric but still be functionally correlated. What was needed is a more expressive metric that can capture probabilistic relations operating on the distributions. One option is the *relative entropy* or *Kullback-Leibler* distance [3] which is a measure of the distance between two probability distributions. One problem with the Kullback-Leibler distance is that it is not symmetric and that it does not satisfy the triangle inequality, which also means that it is not a metric [3]. Instead we use Crutchfield's *information metric* [4] which measures the distance between two information sources.

Let $\mathcal{X}$ be the alphabet of values of a discrete random variable (information source) X with a probability mass function $p(x)$, where $x \in \mathcal{X}$. The conditional entropy

$$H(Y|X) = -\sum_{x \in \mathcal{X}} \sum_{y \in \mathcal{Y}} p(x, y) \log_2 p(y|x) \tag{6}$$

is the uncertainty associated with the discrete random variable Y if we know the value of X. In other words, how much more information do we need to fully predict Y once we know X. The information metric [4] is the sum of these two uncertainties, or formally

$$d(X, Y) = H(X|Y) + H(Y|X). \tag{7}$$

Note that X and Y in our system are information sources whose $H(Y|X)$ and $H(X|Y)$ are estimated from the time series of sensors x_i and x_j using Equation (6).

One from our point of view attractive property of the information metric is that two different sources can be informationally equivalent, producing superficially very different time series.

Note that both this metric and the frequency metric depend on the partitioning of the input into a number of subintervals (bins), and that the results might depend on this partitioning [4]. For our experiments an ad-hoc partitioning was used with 50 uniformly distributed subintervals in the range $[0.0, 1.0]$.

3. Experiments and Results

3.1. Distance Metric Differences

To investigate the differences between the Hamming metric, the frequency metric and the information metric we

used as input to the 100 sensors values from a sine-function, see Figure 1. For each time step t each sensor is assigned $x_i(t) = sin(t+d_i)$, where d_i is chosen so that sensor x_{i+50} is the additive inverse of sensor x_i.

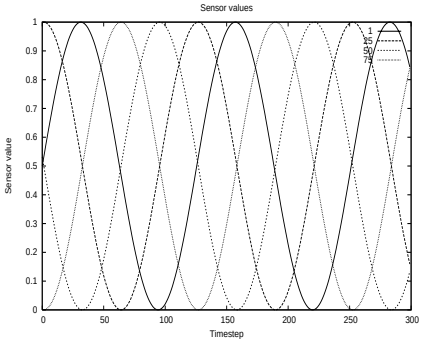
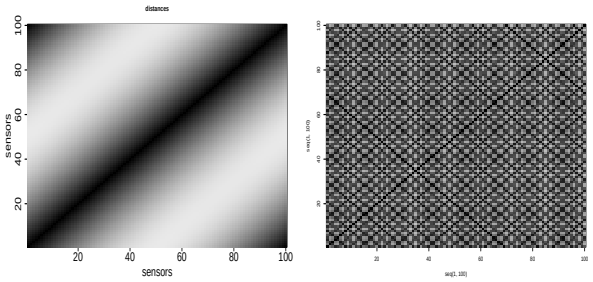
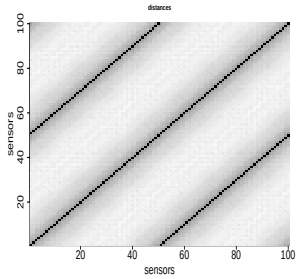


Figure 1. Four of the 100 inputs to the sensors in the *sin* environment. Note that sensor input i always is the additive inverse of sensor input $i + 50$ and vice versa.

Figure 2 shows the distance matrices for the Hamming distance, frequency metric, and the information metric. To begin with one can clearly see the sine-function as the diagonal wave in Figure 2(a) of the Hamming metric. Also, in Figure 2(a) we note that (obviously) the difference between sensor x_i and x_i is 0.0, hence the black diagonal. The further apart the sine-functions are phase-shifted the larger the distance will be. Figure 2(b) shows the frequency metric which does not distinguish well between sensors which have roughly identical frequency distributions. Figure 2(c) shows the distance matrix for the information metric. Here all pairwise distances are almost the same apart from when a sensor is compared with itself or sensor x_i with x_{i+50}, where the distance is 0.0. This shows that the information metric identifies sensor x_i and x_{i+50} as being identical due to their complete correlational relationships over the time series. Also, since the distances between all the sensors is more or less the same, all 100 sensors would be grouped together using the closed subgroup method described by Equation (5) in Section 2. This is different from the Hamming metric where all the sensors would appear in different subgroups.



(a) Hamming metric (b) frequency metric



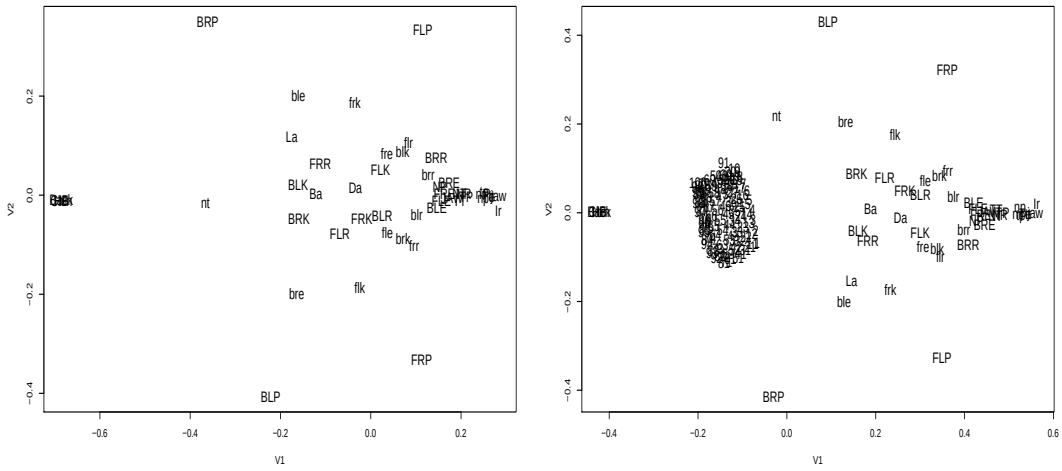
(c) information metric

Figure 2. Distance matrices for the Hamming metric, the frequency metric, and the information metric for the *sine* environment. In all graphs black means a difference of 0.0 and in Figure 2(a) white is 0.7, in Figure 2(b) 0.07, and in Figure 2(c) 3.7.

3.2. Mapping Sensory Organization

Experiments were also performed using data from a SONY AIBO robot chasing a pink ball in an office environment. Sensor data and visual data were collected at an average of 10 frames per second. Since the data was collected over a wireless network the frame rate varied between 8.7 to 11.2 frames per second. A total of 1000 frames were collected. Each frame consists of a 88 x 72 pixels image, 1 distance sensor, 3 acceleration sensors, 1 power sensor, 1 temperature sensor, 8 touch sensors (including 4 paw sensors), 18 joint position values, and 18 motor values. All the input data was normalized to the range $[0.0, 1.0]$ and due to space and time constraints only 10 x 10 pixels taken from the upper left corner of the image was used. Thus the total number of sensors is 150.

Given the 1000 frames of data to the 150 sensors a metric projection can be created of the sensors. Maps for the Hamming metric can be found in Figure 3 and for the information metric in Figure 4. The first thing to notice about both Figure 3 and Figure 4 is the symmetry and organization. In both map Figure 3(b) and Figure 4(b) the visual sensors are clearly grouped together. Even without knowing that sensors 1-100 are vision sensors one could guess that they are sensors positioned in a square formation that measure some kind of piecewise smoothly varying phenomena. There is



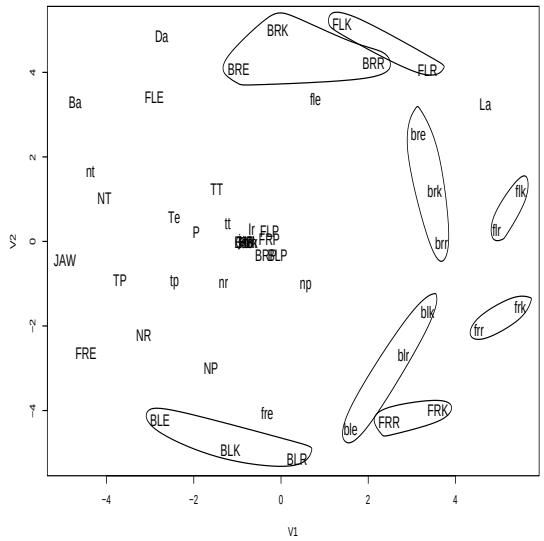
(a) (b)

Figure 3. Metric projections using Hamming metric. Figure 3(a) shows all sensors apart from the 10x10 vision sensors and Figure 3(b) all sensors and the vision sensors. The vision sensors are labelled 1-100, the infrared distance sensor *Ir*, **the acceleration sensors** *Ba* **(y-axis),** *La* **(x-axis), and** *Da* **(z-axis).***Te* **is the temperature sensor and** *P* **the power left in the battery. For the legs** *FL* **means front left,** *FR* **front right,** *BL* **back left, and** *BR* **back right. Each leg has one paw sensor that can be either 1 or 0,** *BLP*, *BRP*, *FLP*, **and** *FRP*. **Each leg has three degrees of freedom, namely elevation** *E*, **rotation** *R*, **and knee** *K*. **Each degree of freedom has a current position, denoted by upper case names, and current duty (motor value), denoted by lower case. For example,** *FLK* **is front left knee position,** *brr* **back right rotation duty, and** *FRE* **front right elevation position.** *NT*, *NP*, **and** *NR* **are neck tilt position, neck pan position, and neck neck roll position respectively. The same codes in lower case are duties.** *T* **and** *t* **means tail and for** *TT*, *TP*, **and** *TR* **the second letter the same as for the neck. Finally,** *JAW* **is the jaw position and** *jaw* **the duty.**

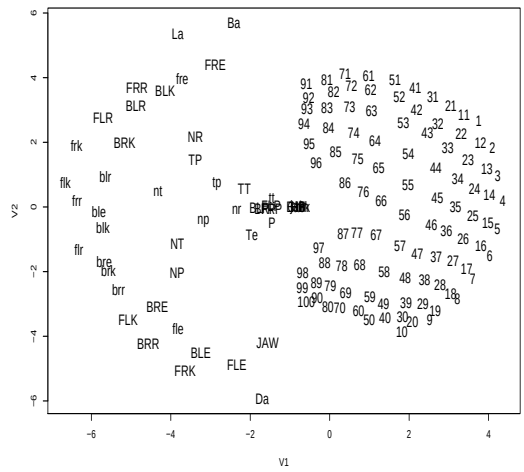
also clear bilateral symmetry along the horizontal axis in all four maps. It is important to note that the positions of the non-visual sensors differ between the map with and without visual sensors. This is because the 100 vision sensors highly affect the positioning of the other sensors in the relaxation algorithm because each sensor applies a force to each other sensor in the algorithm.

Looking at the metric projections in more detail and in particular Figure 3(a) and Figure 4(a) we find the body of the robot reflected in the map. Consider for example Figure 4(a), the information metric applied to the sensory data without the image data. The back left leg positions *BLE*, *BLK*, and *BLR* (see Figure 3 for explanation of the different leg positions) are close together and more or less mirrored by the back right leg positions *BRE*, *BRK*, and *BRR*. Because the robot used a gait optimized for robotic soccer the front legs have different relative positions than the back legs due to the forward leaning gait. The motor duties of the back left (*ble*, *blk*, *blr*) and right (*bre*, *brk*, *brr*) legs are also grouped together. The sensor maps of Figure 3 and Figure 4 can be seen as the first steps toward "AIBO-unculus" maps, in a sense similar to homunculus maps of humans where each region of the body is represented by corresponding regions both in the somatic sensory cortex and primary motor cortex [10].

Now consider the paws *BLP*, *BRP*, *FLP*, and *FRP* and their position in the metric projection generated by the Hamming metric in Figure 3. Here they are positioned in 4 different corners of the map. However, if we look closely in the map for the information metric in Figure 4 we find the paws closely grouped together in the middle of the map. Similarly to the experiment with the sine-functions the paws are grouped together because the functions generating the data are more or less the same, only phase-shifted in time and thus are mutually predictable. This is another example that the information metric can find functional structure not



(a)



(b)

Figure 4. Metric projections using the information metric. Figure 4(a) shows all sensors apart from the 10x10 vision sensors and Figure 4(b) all sensors and the vision sensors. See Figure 3 for an explanation of the labels used in the graphs.

found by the Hamming metric.

There are also sensors that do not seem to be related at all that are very close together in the maps, e.g., the tail and the temperature sensors. In this experiment temperature was roughly constant, and the tail was never used, so its position and motor values were static and thus the information distance is very close to 0.0. By performing active probing we expect it to be possible to separate sensors that have been closely grouped together.

3.3. De-Scrambling of Vision Sensors

Finally we present an application of the method. Again consider Gregor Samsa who awoke in the body of a monstrous bug and has to interpret a deluge of bizarre sensory stimulation. Similarly we can imagine a new control system placed inside the body of a Sony AIBO robot. When the camera of the AIBO is connected to the control system all the wires are mixed up and the output from the vision system looks like Figure 5(a). But is this really noise or not?

Imagine that each pixel in the camera is connected with a wire to the corresponding pixel location in the control system. The image will then look like Figure 5(b), which is what the camera sees. But, if the wires are connected to random pixels in the control system the same input to the camera will look like Figure 5(a) in the control system. This information-preserving random mapping can be found if we assume that either the AIBO is moving around in the world or that some objects in the visual field of the AIBO are moving. By creating a metric projection using the Hamming meric of only the vision sensors we get a projection like Figure 5(e). This projection is an estimate of how the sensors are connected to the positions in the control system from the camera. In this case we are using a 20 x 20 pixel image with sensor 1 in the eye place in the upper left corner and sensor 400 in the lower left corner. The projection shows in what position for example sensor 42 should be positioned in to get the original image.

To go from the continuous projection of Figure 5(e) to a two-dimensional image we need an algorithm that as input take the the x and y positions of the n sensors in the projection, and as output these points mapped to a discrete (in this case) 20 x 20 grid. To do this we have developed a quite trivial algorithm that, as it has turned out, is quite effective. The algorithm works as follows. Order all points according to their position along the x-axis in the projection with the smallest x-axis value first and add them to the list L_{all}. Pick the first $\sqrt{n}$ points from L_{all} and order them according to their position along the y-axis in to list $column_k$, where k is counter that is incremented by 1 each $\sqrt{n}$ points picked from L_{all}. Then delete the points in $column_k$ from L_{all}. Repeat until L_{all} is empty. Now each list $column_k$ contains the points for one column of the image ordered by k. This assumes that the layout is square. For rectangular layouts first the number of columns and rows must be found.

One example of a de-scrambled 20 x 20 image taken

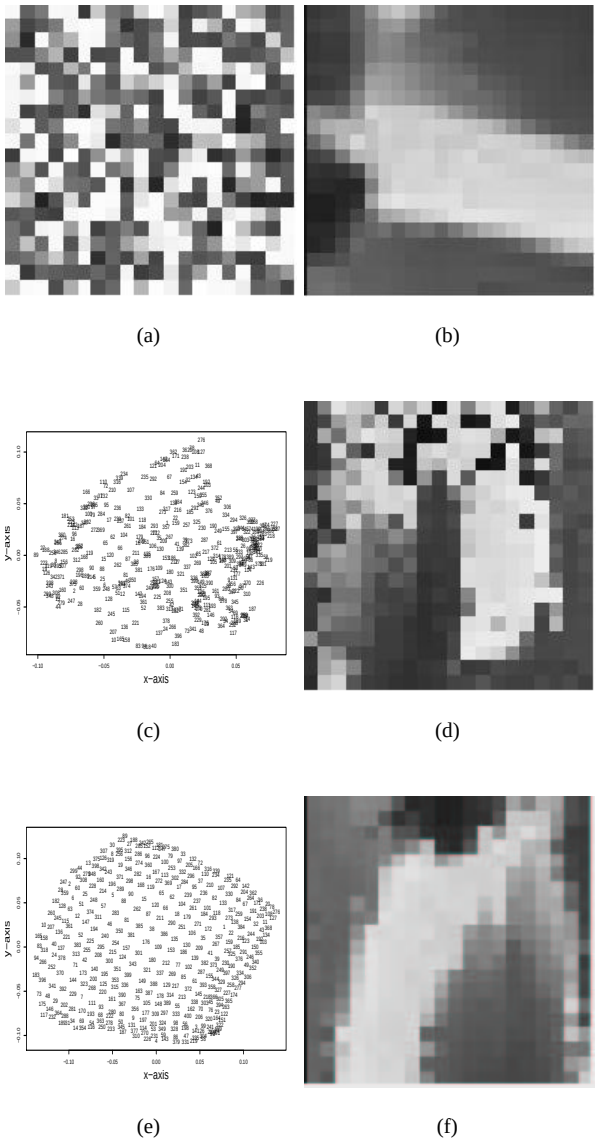


Figure 5. Figure 5(a) is one image from input with scrambled sensors (20 x 20 pixels) and Figure 5(b) shows the original image. Figure 5(c) shows the metric projection and Figure 5(d) the de-scrambled image after 100 frames. Figure 5(e) shows the metric projection and Figure 5(f) the image after 1000 frames of input data.

from a SONY AIBO is Figure 5(d) which is the de-scrambling attempt of the scrambled image of Figure 5(a) after 100 frames (time steps) of input data. As we can see some structure is starting to appear in the image. Figure 5(f) shows the same image after 1000 frames of input data. Here the resulting image is rotated 90 degrees clockwise. This is due to the fact that the sensory information coming in from the vision contains no directional information, and only the relative positions between the sensors can be computed. Thus, the resulting random mapping can have 8 possible orientations with the same probability. To achieve better quality a higher frame rate and more data is needed.

The reason that this works is the smooth effects of movement (either by the sensors or objects in front of the sensors) of the sensory features. A vision system that is not moving and has no moving objects in front of it will not be able to de-scramble the random mapping.

4. Conclusions and Future Work

We have in this paper extended the work in [15] by introducing the information metric [4] as an additional distance metric. A theoretical experiment using sine-functions as input to the sensors show that the information metric can capture informational structure not found by the distance metrics described in [15]. Metric projections computed using the different distance matrices can reveal interesting properties of the sensory data if the structure and nature of the data is not known. Data from a SONY AIBO robot walking around in an office environment has been used to create metric projections of the AIBO's sensors. These projections are symmetrical and contain information about the functional layout of the sensors. For example, it is possible to find the vision sensors and their relative physical ordering from raw uninterpreted data from all sensors. It is also possible to reconstruct functional relationships of parts of the physical body by studying the relations between sensors and related motor values. Finally we showed an application of the metric projections where the real layout of vision sensors is recovered from scrambled input data.

It is important to note that the actions of the agent to some extent determine the layout of the metric projections - what you do in the world determines what you can distinguish.

In addition to the application of the method described in this paper to self-configuring and evolving robots and hardware, we see another potential application area in sensor networks [1]. Much work in sensor networks has focused on difficult technical issues like constructing the hardware, ad-hoc networks, and the design of protocol layers. But, what happens when all of these potentially thousands or millions of smart sensors are connected together? How will the data be integrated and interpreted, especially if nodes disappear and then maybe reappear again in another position? Here it might be possible to apply the methods described in our paper. For instance, consider 100 sensors in a sensor network that each have a tiny camera. By creating metric projections of their input one can possibly integrate the data from these

sensor nodes into one global view. If better resolution is needed 50 more sensor nodes can be added and integrated to the global image. It is also possible to identify areas of the sensor network where the sensors transmit redundant data or where more sensors are needed.

References

[1] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci. A survey on sensor networks. *IEEE Communications Magazine*, 40:102–114, 2002.

[2] P. Cariani. Epistemic autonomy through adaptive sensing. In *Proceedings of the 1998 IEEE ISIC/CRA/ISAS Joint Conference*, pages 718–723, 1998.

[3] T. M. Cover and J. A. Thomas. *Elements of Information Theory*. John Wiley & Sons, Inc., N. Y., 1991.

[4] J. P. Crutchfield. Information and its Metric. In L. Lam and H. C. Morris, editors, *Nonlinear Structures in Physical Systems – Pattern Formation, Chaos and Waves*, pages 119–130. Springer Verlag, 1990.

[5] K. Dautenhahn, D. Polani, and T. Uthmann. *Artificial Life (Special Issue on Sensor Evolution)*, 7(2), 2001.

[6] D. B. Dusenbery. *Sensory Ecology*. WH Friedman & Co, first edition, 1992.

[7] G. Goodhill, S. Finch, and T. Sejnowski. Quantifying neighbourhood preservation in topographic mappings, 1995.

[8] T. Higuchi, M. Iwata, D. Keymeulen, H. Sakanashi, M. Murakawa, I. Kajitani, E. Takahashi, K. Toda, N. Salami, N. Kajihara, and N. Otsu. Real-world applications of analog and digital evolvable hardware. *IEEE Transactions on Evolutionary Computation*, 3(3):220–235, Sep 1999.

[9] F. Kafka. *The Transformation (Metamorphosis) and Other Stories*. Penguin Books, first edition, 1992.

[10] E. Kandel, J. Schwartz, and T. Jessell. *Principles of Neuroscience*. McGraw-Hill, fourth edition, 2000.

[11] D. Keymeulen, R. Zebulum, A. Stoica, and M. Buehler. Initial experiments of reconfigurable sensor adapted by evolution. In *International Conference on Evolvable Systems, October 2001*, pages 303–313, 2001.

[12] W. J. Krzanowski. *Principles of Multivariate Analysis: A User's Perspective*. Clarendon Press, Oxford, 1988.

[13] N. Macias and L. Durbeck. Self-assembling circuits with autonomous fault handling. *Proceedings of the Fourth NASA/DoD Workshop on Evolvable Hardware, July 2002*, 2002.

[14] C. L. Nehaniv. Meaning for observers and agents. In *IEEE International Symposium on Intelligent Control / Intelligent Systems and Semiotics*, pages 435–440. IEEE Press, 1999.

[15] D. Pierce and B. Kuipers. Map learning with uninterpreted sensors and effectors. *Artificial Intelligence*, 92:169–229, 1997.

[16] D. Polani. Measures for the organization of self-organizing maps. *Self-Organizing neural networks: recent advances and applications*, pages 13–44, 2002.

[17] D. Polani, T. Martinetz, and J. Kim. An information-theoretic approach for the quantification of relevance. In *Advances in Artficial Life (Proc. European Conf. Artificial Life - ECAL'01), Springer LNAI vol. 2159*, pages 704–713, 2001.

[18] C. E. Shannon. A mathematical theory of communication. *Bell System Tech. J.*, 27:379–423, 623–656, 1948.